

# Build User Daily Activity Model and Model Structure Changing

GuoQing Yin and Dietmar Bruckner

Institute of Computer Technology  
Vienna University of Technology, Austria, Europe  
{yin, bruckner}@ict.tuwien.ac.at

**Abstract-** because of the activity dynamic it is a challenged work to build daily activity model of user. The goal of the paper is to build daily activity model of the user with high accuracy. At first the raw data derived from motion detector will be “translated” to state data, then use state split and merge to build the basic model. Thirdly in order to increase accuracy of the model the count of the state data increased by changing the parameter of the translator. Here we get another two activity models with higher accuracy.

The changing of the count of the state data caused to the structure changing of the build activity model. We merge the states from same route of activity models, compare different models and find out the same activity trends of the user. On the other hand from the models with higher accuracy the hidden activity will be detected.

**Keywords:** activity model, ambient assisted living, structure changing, structure learning.

## I. INTRODUCTION

Many papers about activity model and structure learning: in paper [1] the authors provide a computationally efficient method for learning Markov network structure from data. In case of variables in real-world systems often group into classes that predict the kinds of probabilistic dependencies they participate in the authors of paper [2] exploit the prior knowledge to enable structure learning. The authors of paper [3] develop an algorithm for learning the structure of Markov logic networks from relational databases, combining ideas from inductive logic programming and feature induction in Markov networks. In paper [4] the authors review the evidence for structure learning as a “learning to learn” mechanism and utilize it to sensorimotor control. The authors of paper [5] proposed implicit score to learn Bayesian network structure from database. The authors of paper [6] design a generic fitness function for validation of input specification and use it to develop genetic algorithm wrappers. One of the wrappers used to deal with the variable ordering problem for Bayesian network structure learning. In paper [7] and [8] the authors introduced states merging method that use as security, care system, and building automation system.

### A. AAL, ATTEND Project

Ambient Assisted Living (AAL) are aim to improve the life quality of the elderly with the help from modern technology and prolong the independent living in their own home. But

the elderly have their own problems because of aging, such as action obstacles, memory disorder..., how can the modern technology system adapt to the elderly?

A system will be developed at the project ATTEND (AdapTive scenario recogniTion for Emergency and Need Detection) that increases the independent living of elderly. Through an intelligent, adaptive network of sensors which installed in the living environment, the activity information of the user will be collected. Then the user’s activities and behavior model will be learned and build by the system after a time period. According the model if there are unusual activity happened by the user an alarm signal will be send to the caregiver.

### B. Contribution

In earlier paper [9] Gaussian mixture model and split-merge algorithm be used to analyze the sensor data and built the statistical activity model of the user. In paper [10] hidden Markov model, the forward algorithm, and the Viterbi Algorithm be used to build the person’s daily activity model. This paper presents the whole process from raw sensor data to build activity model of the user and further more discussed the model structure changing. A filter algorithm use to transfer sensor raw data to state data. States split and merge use to build activity model of the user and simplify the build model. In order to get higher accuracy model the parameter of the filter algorithm will be changed. The parameter changing caused to model structure changing will be discussed. The result showed that the hidden activity will be detected from the higher accuracy model.

## II. TRANSLATE RAW SENSOR DATA TO STATE DATA

The basic function of the motion detector is that: the motion detector sensor sends the value “1” to the controller if there are activities from the user. Otherwise the motion detector sends the value “0” to the controller. For continuous activities or stillness the sensor value keeps on “1” or “0”.

The problem of the activity of the user is that they are dynamically. For example we don’t know what the elderly will do in the next second (or minutes) in his daily living. The activities of the user are dynamically. But some of the elderly have stable living habit. For example when gets up in the morning, takes a shower... goes to bed in the evening. These kinds of activities happen every day and at the nearly same

time. So according the living habit of the elderly we can say: what time what kind of activity the user will do generally. This is the basic idea in the paper the activity model to build. But the problem is that in real situation the sensor (install in the living room) raw data are random distributed and with huge quantity in each day. To deal with individual data has no sense to build an activity model, but if gather individual data together in some time intervals, the generally activity information of the user will be indicated [10]. For example 60 minutes as a time interval, so there will be 24 time intervals in one day (for more detail of the activity of the user the time interval should be shorter but at the same time the count of the time intervals will be increased, this point will be explained in the following with examples). Then gather the activities of the sensor in each time interval. If the sum values of activities bigger than a predefined threshold value, so the activity value of the time interval will be treated as "1", others have value "0". Here the predefined threshold value and the time interval play an important role.

- The gathered sensor value (for example in 60 minutes)

$$T = \{t_{1(1)}, t_{2(0)}, t_{3(1)}, t_{4(0)}, t_{5(1)}, t_{6(0)} \dots t_{n(v)}\} \quad (1)$$

Here  $t_n$  is the time point that the motion detector send value to controller ( $n \geq 1$ ).  $V$  is the sensor value itself, it has value "0" or "1".

- The activities between sensor value

$$\Delta T = (t_{n(0)} - t_{n-1(1)}) \quad (2)$$

- Sum all the activities in a time interval (for example in 60 minutes)

$$T' = \sum(\Delta T) \quad (3)$$

- Deciding the time interval gets value "1" or "0".

$$\text{If } T' \geq T_{th} * T_{interval}, \text{ then } Q_{interval, ix} = 1 \quad (4)$$

$$\text{If } T' < T_{th} * T_{interval}, \text{ then } Q_{interval, ix} = 0 \quad (5)$$

Here  $T_{th}$  is the threshold value,  $0 < T_{th} < 1$ .  $T_{interval}$  is the predefined time interval.  $Q_{interval, ix}$  is the result value that the interval should take. Here "ix" is the interval count (index).

If we chose 60 minutes as  $T_{interval}$  (so we get 24 state data from hundreds raw data) and 0.2 as  $T_{th}$  to filter the original sensor data for one observed day, we can get state data sequences, such as: [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1]. Here are totally 24 value (states), each value indicated the state of the corresponding time interval. If the value is "0", that means in the time interval without activities (or the activity in a low level) and the value "1" means that there have high activities in the time interval.

From above 24 states value we can deduce that from 1 to 8 o'clock the user has lower activity (at night in sleeping situation) at 9 o'clock there has activity from the user then at 10 o'clock with low activity. From 11 to 16 o'clock the user has continuous activity then from 17 to 18 there are low activity and from 19 to 24 o'clock there are activity from the user. From these states value we get the generally activity information of the user in one day. If we gather the data for a longer time interval, for example one week, so we can get a more completely activity states model of the user. Now another question emerges: how build these sates from different days to one model?

### III. SPLIT AND MERGE OF STATES ROUTES

To build states route from different days to one activity model the states split and merge needed. In the following the rule will be explained with graphic.

#### A. States Split

For example there are 2 state routes from different days (on top of figure. 1). At start the states have same value (0) then have different value (the top state route has continuous value 0 but the bottom state route change state value from 0 to 1). In such situation the states with same value will be merged together and from the states with different value the states route will be split. The bottom of figure 1 shows the state route split to 2 different routes.

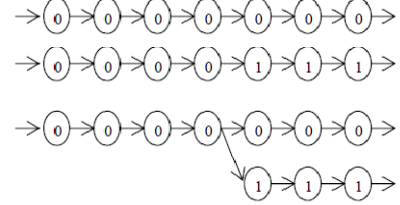


Fig. 1. The state route split from the states with different value

#### B. States Merge

States merge (figure. 2) is on the contrary of states split. At start the states have different values (the top state route has continuous value 0 but the bottom state route change state value from 1 to 0) then have same value (0). In such situation the states with different value will stay at different state route and from the states with same value the states route will be merged together. The bottom of figure 2 shows the state routes merge to 1 state route.

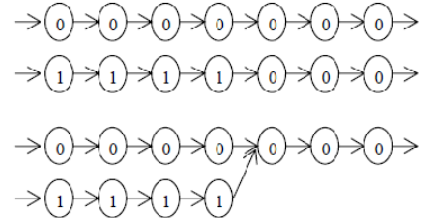


Fig. 2. The state route merged from the states with same value

#### C. The Completely Activity Model

Through above explanation (states split and states merge) the completely activity model of the user will be build with the introduced method. In figure 3 the x-axis is the state value the y-axis is the count of the day. Figure 3 shows that all 7 days activity state routes composed to one model (the raw data from a motion detector installed in living room). Each small "." represents one state, has state value "0" or "1". The first state (on the left side of day 7) is the initial state, it has without state value. The last state (on the right side of day 7) is the end state, it has without state value. The green "circle" means states with value "0" and the blue "square" means state value "1". The activity of the user in one week will be totally expressed in the activity state model. We get the generally information about the activity of the user. But in figure 3 there are some states have the same state value and they are

consecutive. Through state merge these states can merge again, so the activity model will be simplified further.

From figure 3 the generally activity model indicated that the user has lower activity from 0 to 10 o'clock, from 11 to 16 there are more activity, then from 17 to 24 o'clock a few days with low activity and other days with high activity.

#### D. Merge the States with Same Value

In figure 3 there are some consecutive states with the same state value (or different value alternately), if we merge these states together, so the states in the model will be reduced. It helps us to get one more simplify and generally state model. The merge method will be introduced in figure 4 and 5.

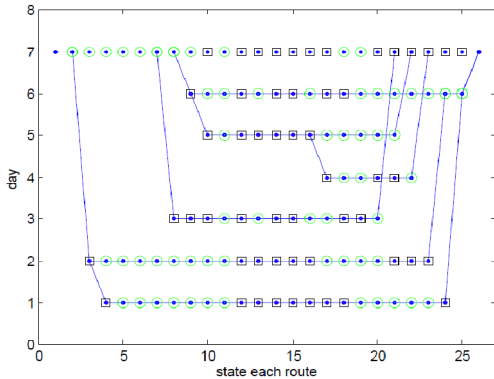


Fig. 3. The activity model after state split and merges in one week

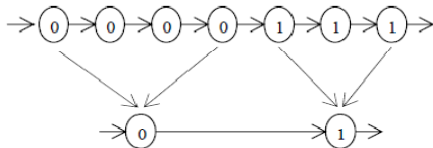


Fig. 4. Merge the consecutive states with the same value

In top of figure 4 there is a state route with consecutive state value “0” and “1”. These consecutive states will be merged to one state with the same state value. In bottom of figure 4 indicated the merged result. 4 states with value “0” merged to one state with same value and 3 states with value “1” merged to one state with the same value “1”.

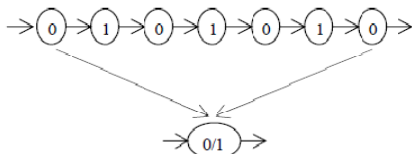


Fig. 5. Merge the consecutive states with the alternately value

In top of figure 5 there is a state route with consecutive state value “0” and “1” alternately. These consecutive states will be merged to one state. In bottom of figure 5 indicated the merged result. The 7 states with alternately state value “0” and “1” will be merged to one state and the state has mixture value “0/1”. After merging the states of the model will be reduced significantly. Figure 6 shows the result.

In figure 6 the x-axis is the state count and y-axis is the count of day. The state with green circle means that the state

has value “0”. The state with black square means that the state has value “1” and state with blue star means that the state includes value “0/1”. The blue star in a green circle that means the first state is “0” and if the blue star in the black square that means the first state with value “1”. Comparing figure 3 and 6 it is clearly that the states count reduced significantly.

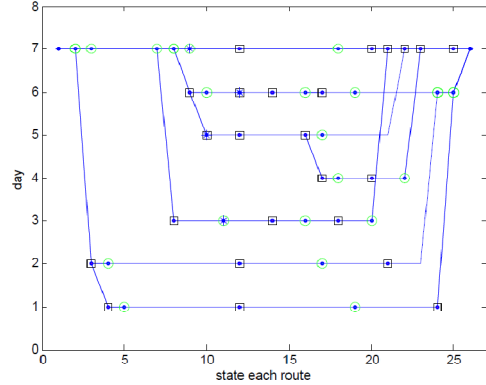


Fig. 6. The activity model after state merging

Figure 6 indicated the activity of the user on every day in one week. But the problem is that the chosen time interval ( $T_{interval}$ ) of each state is one hour, for some short time duration activity of the user the chosen  $T_{interval}$  is long. That means a chosen  $T_{interval}$  with bigger value caused to a generally activity model but not accurate enough. In order to search the activity more in detail we try to change the parameter  $T_{interval}$  smaller, for example 30 or 15 minutes as  $T_{interval}$ , that cause to the states count increasing and further more the whole activity model structure changing. The point will be discussed following.

#### IV. MODEL STRUCTURE CHANGING

If we change the parameter  $T_{interval}$  smaller the count of the states in a day will be created. For example if the  $T_{interval}$  is chosen 60 minutes, there should be 24 states in a day, but if  $T_{interval}$  change to 30 (or 15) minutes, so there will be 48 (or 96) states in a day. With more states the accuracy of the user activity will be increased. Figure 7 illustrates that changing the parameter  $T_{interval}$  cause to state count changing.

On top of figure 7 is the states with  $T_{interval}$  60 minutes, on the middle of figure 7 there 30 minutes as  $T_{interval}$  for each state, on the bottom of figure 7 each state has  $T_{interval}$  value 15 minutes. It is clearly the count of the states increased double when  $T_{interval}$  reduced half. Through this kind of changing the dynamic situation of the user will be fixed on state value with more accuracy. For example at the third state on the top of the figure 7 the state value is “0”, that means in whole one hour the user activity is low. When the state split to 2 states on the middle of figure 7 they have value “1” and “0”. That means the activity of the user at the first 30 minutes with value “1” and the later 30 minutes with value “0”. Furthermore the 2 states split again, on the bottom of the figure 7 there are 4 states with value “1, 0, 0, 0”, that means at the first 15 minutes the activity of the user with value “1” and the second,

third and fourth 15 minutes with lower activity. So the dynamic situation of the user from “one hour lower activity” step to step change to “30 minutes higher activity, 30 minutes lower activity”, then “15 minutes higher activity and 45 minutes lower activity”. The accuracy of the activity increased 4 times. More accuracy of the activity helps us to aware the activity situation of the user more in detail. On the other side if the parameter  $T_{interval}$  be chosen too small, for example 1 or 2 minutes, there will be 1440 (or 720) states in a day. The state data will be as much as the sensor raw data, the activity situation will be trap into detail and the build model will be too complicated. In our opinion the parameter  $T_{interval}$  be chosen as 10 or 15 minutes is appropriate.

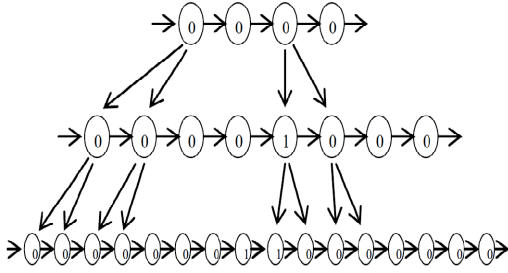


Fig. 7. The state count changing by  $T_{interval}$  changing

Above we discussed how the parameter  $T_{interval}$  changing cause to the state count and value changing, furthermore the changing also cause to the model structure changing. This point will be discussed following.

Because of the state count and value changed so the states merge and split possibly happen to different state (figure 1 and 2). This directly caused to the whole model structure changing. Comparing figure 3 (with 24 states each route) to figure 8 (with 48 states each route) the model structure changed. And furthermore the structure changed again in figure 9 (with 96 states each route).

Although the model structure changing the mainly activity information are similarly but with more accuracy.

From figure 8 and 9 we get more information that has not clearly indicated in figure 3. In all 7 days there are 4 nights that the user has activity from 30 minutes to 90 minutes (in figure 3 there are 2 nights the user has activity with 1 hour). In figure 9 there are 5 nights that the user has activity from 15 minutes to 30 minutes. Especially at night on day 3 there are alternatively activity from the user. It indicated a sleeping disorder possibly.

Furthermore these states in figure 8 and 9 can be merged again, the simplified model showed in figure 10 and 11. Comparing figure 10 to 8, the states count reduced significantly but included the same information. Comparing figure 10 to 6, the model structure changed and figure 10 included information with more accuracy.

In figure 11 the parameter  $T_{interval}$  is 15 minutes. Comparing figure 11 to 9, the states count reduced significantly but included the same information. Comparing figure 11 to 10 and 6, the model structure changed and figure 11 included information with more accuracy than figure 10 and 6.

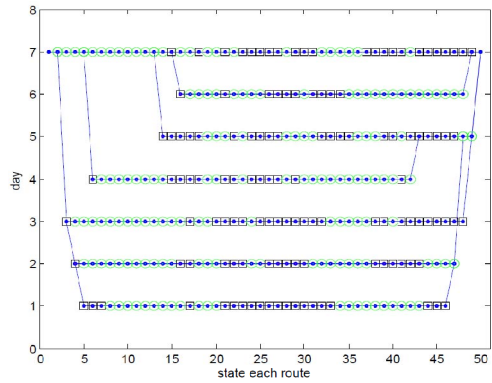


Fig. 8. The states model with 30 minutes as  $T_{interval}$

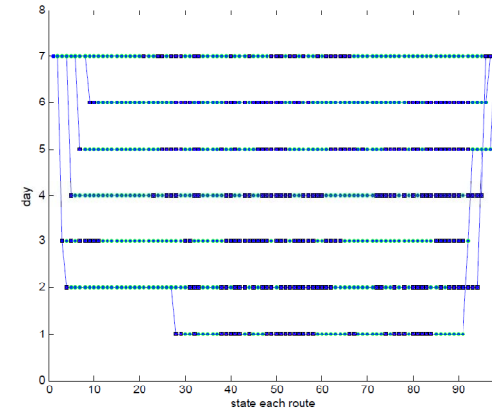


Fig. 9. The states model with 15 minutes as  $T_{interval}$

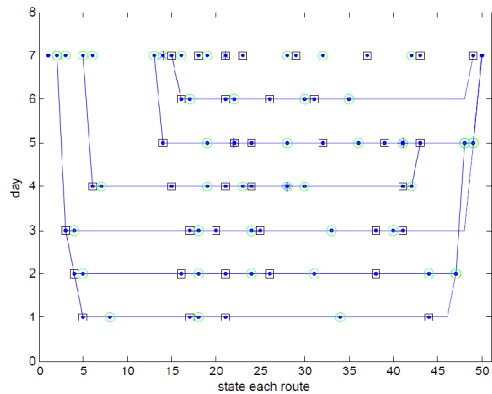


Fig. 10. The merged states model with 30 minutes as  $T_{interval}$

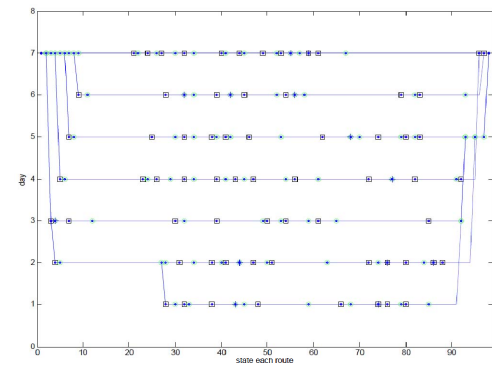


Fig. 11. The merged states model with 15 minutes as  $T_{interval}$

Figure 12 shows the state count changing because of different  $T_{interval}$ . The line in top of figure 12 indicates the original state count in all 7 days with 15, 30 and 60 minutes as  $T_{interval}$ . The dash line in figure 12 shows the state count after state split and merge (corresponding figure 3, 8 and 9) and the point line shows the state count after consecutive state merge (corresponding figure 6, 10 and 11). The slopes of the top and middle lines are similarity but different to the point line. That means state split and merge don't change the state count character of the model so much. The consecutive state merge indicated the kernel activity model of the user. It doesn't change too much in spite of  $T_{interval}$  and the model structure changing.

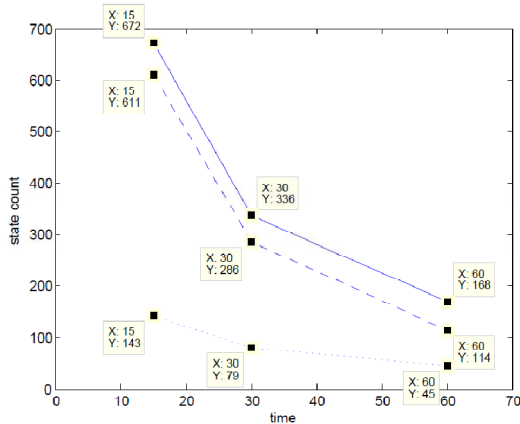


Fig. 12. Comparing the original and merged states with different  $T_{interval}$

## V. CONCLUSION

The paper introduced a filter algorithm that use to transfer sensor raw data to state data, then discussed states split and merge in order to build activity model of the user and simplify the build model. Through changing the parameter  $T_{interval}$  of the filter algorithm the structure of the model changed and the accuracy of the model increased. With the higher accuracy model the hidden activity will be detected (The result showed in figure 3, 8, and 9. For example in figure 3 on day 3 there are without activities from the user at night, but in figure 8 on day 3 one state indicates activities from the user at night, and in figure 9 on day 3 many states indicates activities from the user at night).

Although the parameter changing, model structure changing the mainly activity information of the used are similarity. It indicates that the stable life style of the user is the basic of a relatively stable activity model.

## VI. OUTLOOK

In the paper we introduced to build activity model of the user in one week as example. For a more flexible and tolerant model the example days should be more, for example one month or longer. So there should be about 30 routes in the model and the model structure will be complicated. The next step work is that try to find a method to merge the route in different days in order to reduce the routes.

Furthermore should according the build model to find the most adapt route compare to an observation route and try to find out abnormal activity of the user.

Because the model is just from single motion detector, it is difficult according the model to distinguish between humans, animals or objects. In order to realize such goal sensor fusion should be utilized. The work will be done in the next step.

## REFERENCES

- [1] Su-In Lee, Varun Ganapathi, and Daphne Koller, "Efficient Structure Learning of Markov Networks using  $L_1$ -Regularization", Department of Computer Science, Stanford University.
- [2] V.K. Mansinghka, C. Kemp, J. B. Tenenbaum, and T. L. Griffiths, "Structured Priors for Structure Learning", Dept. of Brain & Cognitive Sciences Massachusetts Institute of Technology Cambridge, MA 02139 and Dept. of Cognitive & Linguistic Sciences Brown University Providence, RI 02912.
- [3] Stanley Kok and Pedro Domingos, "Learning the Structure of Markov Logic Networks", in *Proceedings of the 22 nd International Conference on Machine Learning*, Bonn, Germany, 2005.
- [4] Daniel A. Braun, Carsten Mehring, and Daniel M. Wolpert, "Structure learning in action", *Behavioural Brain Research* 206 (2010) 157–165.
- [5] Lobna Bouchaala, Afif Masmoudi, Faiez Gargouri, and Ahmed Rebai, "Improving algorithms for structure learning in Bayesian Networks using a new implicit score", *Expert Systems with Applications* 37 (2010) 5470–5475.
- [6] William H. Hsu, "Genetic wrappers for feature selection in decision tree induction and variable ordering in Bayesian network structure learning", *Information Sciences* 163 (2004) 103–122.
- [7] Dietmar Bruckner, Brian Sallans, and Gerhard Russ, "Probability Construction of Symbols in Building Automation Systems", in: *Proceedings of 2006 IEEE International Conference of Industrial Informatics INDIN' 06*, S. 6, Singapore, 2006.
- [8] Dietmar Bruckner, Brian Sallans, and Roland Lang, "Behavior Learning via State Chains from Motion Detector Sensors", *Biometrics '07* December 10-13, 2007, Budapest, Hungary.
- [9] G. Yin, D. Bruckner: Split-Merge Algorithm and Gaussian Mixture Models for AAL. In *IEEE International Symposium on Industrial Electronics (ISIE 2010)*, Bari, Italy, July 4-7, 2010.
- [10] G. Yin, D. Bruckner: Daily Activity Learning from Motion Detector Data for Ambient Assisted Living. In: *Proceedings of the 3rd International Conference on Human System Interaction*, S. 6, Rzeszow, May 13-15, 2010.